AMENDMENTS TO THE CLAIMS:

Please amend claims 8, 43, 44 and 48 as follows.

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. *(Original)* An integrated circuit comprising:

a plurality of processing stages, at least one of said processing stages having processing logic operable to perform a processing operation upon at least one processing stage input value to generate a processing logic output signal; and

a low power mode controller operable to control said integrated circuit to switch between an operational mode in which said integrated circuit performs said processing operations and a standby mode in which said integrated circuit retains signals values but does not perform said processing operations; wherein

said at least one of said processing stages has:

a non-delayed latch operable to capture a non-delayed value of said processing logic output signal at a non-delayed capture time; and

a delayed latch operable during said operational mode to capture a delayed value of said processing logic output signal at a delayed capture time, said delayed capture time being later than said non-delayed capture time, said non-delayed value being passed as a processing stage input value to a following processing stage before said delayed capture time and a difference between said non-delayed value and said delayed value being

indicative of said processing operation not being complete at said non-delayed capture time;

said delayed latch is operable during said standby mode to retain said delayed value whilst said non-delayed latch is powered down and is susceptible to loss of said non-delayed value; and

said delayed latch is formed to have a lower static power consumption.

2. *(Original)* An integrated circuit as claimed claim 1, wherein said delayed latch has a lower speed of operation than said non-delayed latch.

3. *(Original)* An integrated circuit as claimed in claim 1, wherein upon switching from said standby mode to said operational mode said delayed value stored within said delayed latch is passed to as said processing stage input value to said following processing stage.

4. *(Original)* An integrated circuit as claimed in claim 3, wherein said delayed value is copied to said non-delayed latch to said delayed latch upon switching from said standby mode to said operational mode.

5. *(Original)* An integrated circuit as claimed in claim 1, wherein said last least one of said processing stages has:

a comparator operable to compare said non-delayed value and said delayed value to detect a change in said processing logic output signal following said non-delayed capture time indicative of said processing logic not having finished said processing operation at said non-delayed capture time; and

error-repair logic operable when said comparator detects said change to perform an error-recovery operation suppressing use of said non-delayed value by said following processing stage.

6. *(Original)* An integrated circuit as claimed in claim 1, comprising a meta-stability detector operable to detect meta-stability in said non-delayed value and trigger said error-repair logic to suppress use of said non-delayed value if found to be meta-stable.

7. *(Original)* An integrated circuit as claimed in claim 1, wherein when said comparator detects said change, said error-repair logic is operable to replace said non-delayed value with said delayed value as said processing stage output signal.

8. *(Currently Amended)* An integrated circuit as claimed in claim ~~6~~7, wherein supply of said delayed value to said following processing stage forces forward progress through processing operations.

9. *(Original)* An integrated circuit as claimed in claim 1, wherein when said comparator detects said change said error-repair logic is operable to force said delayed value to be stored in said non-delayed latch in place of said non-delayed value.

10. *(Original)* An integrated circuit as claimed in claim 1, wherein processing operations within said processing stage and said following processing stage are driven by a non-delayed clock signal.

11. *(Original)* An integrated circuit as claimed in claim 10, wherein when said comparator detects said change said error-recovery logic is operable to gate said non-delayed clock signal to provide time for said following processing stage to recover from input of said non-delayed value and instead use said delayed value.

12. *(Original)* An integrated circuit as claimed in claim 11, wherein said non-delayed capture time is derived from a predetermined phase point of said non-delayed clock signal, a phased delayed version of said non-delayed clock signal is used as a delayed clock signal and said delayed capture time is derived from a predetermined phase point of said delayed clock signal.

13. *(Original)* An integrated circuit as claimed in claim 1, wherein said plurality of processing stages are respective pipeline stages within a synchronous pipeline.

14. *(Original)* An integrated circuit as claimed in claim 1, wherein a minimum processing time taken for said processing operation is greater than a time separating said delayed capture time from said non-delayed capture time such that said delayed value is not influenced by a processing operation performed upon different input values.

15. *(Original)* An integrated circuit as claimed in claim 14, wherein said processing logic includes one or more delay elements to ensure said minimum processing time is exceeded.

16. *(Original)* An integrated circuit as claimed in claim 1, wherein a maximum processing time taken for said processing operation is less than a sum of a time separating said delayed capture time from said non-delayed capture time and a time between non-delayed capture times such that said processing logic will have completed said processing operation by said delayed capture time.

17. *(Original)* An integrated circuit as claimed in claim 1, wherein said processing stages are part of a data processor.

18. *(Original)* An integrated circuit as claimed in claim 5, comprising an error counter circuit operable to store a count of detection of errors corresponding to said change.

19. *(Original)* An integrated circuit as claimed in claim 18, wherein said count may be read by software.

20. *(Original)* An integrated circuit as claimed in claim 5, comprising a performance monitoring module operable to monitor work quantities including a quantity of useful work performed in progressing said processing operation and a quantity of work used to perform said error-recovery operations.

21. *(Original)* An integrated circuit as claimed in claim 20, wherein one or more operating parameters are controlled in dependence upon said work quantities.

22. *(Original)* An integrated circuit as claimed in claim 21, wherein said one or more operating parameters include at least one of:

an operating voltage;

an operating frequency;

an integrated circuit body bias voltage; and

temperature.

23. *(Original)* An integrated circuit as claimed in claim 1, wherein said delayed latch also serves as a serial scan chain latch within a serial scan chain.

24. *(Original)* An integrated circuit as claimed in claim 9, wherein said delayed latch also serves as a serial scan chain latch within a serial scan chain and a signal value serially scanned in to said serial scan chain latch is forced in to said non-delayed latch during diagnostic operations using said error repair logic.

25. *(Original)* A method of operating an integrated circuit having a plurality of processing stages, at least one of said processing stages having processing logic operable to perform a processing operation upon at least one processing stage input value to generate a processing logic output signal, said method comprising the steps of:

controlling said integrated circuit to switch between an operational mode in which said integrated circuit performs said processing operations and a standby mode in which said integrated circuit retains signals values but does not perform said processing operations;

within said at least one of said processing stages:

capturing in a non-delayed latch a non-delayed value of said processing logic output signal at a non-delayed capture time; and

during said operational mode, capturing in a delayed latch a delayed value of said processing logic output signal at a delayed capture time, said delayed capture time being later than said non-delayed capture time, said non-delayed value being passed as a processing stage input value to a following processing stage before said delayed capture time and a difference between said non-delayed value and said delayed value being indicative of said processing operation not being complete at said non-delayed capture time; and

during said standby mode retaining said delayed value within said delayed latch whilst said non-delayed latch is powered down and is susceptible to loss of said non-delayed value; wherein

said delayed latch is formed to have a lower static power consumption than said non-delayed latch.

26. *(Original)* A method as claimed in claim 25, wherein said delayed latch has a lower speed of operation than said non-delayed latch.

27. *(Original)* A method as claimed in claim 25, wherein upon switching from said standby mode to said operational mode said delayed value stored within said delayed latch is passed to as said processing stage input value to said following processing stage.

28. *(Original)* A method as claimed in claim 27, wherein said delayed value is copied to said non-delayed latch upon switching from said standby mode to said operational mode.

29. *(Original)* A method as claimed in claim 25, comprising within said last least one of said processing stages comparing said non-delayed value and said delayed value to detect a change in said processing logic output signal following said non-delayed capture time indicative of said processing logic not having finished said processing operation at said non-delayed capture time; and

upon detection of said change performing an error-recovery operation suppressing use of said non-delayed value by said following processing stage.

30. *(Original)* A method as claimed in claim 25, comprising detecting meta-stability in said non-delayed value and triggering suppression of use of said non-delayed value if found to be meta-stable.

31. *(Original)* A method as claimed in claim 25, wherein upon detection of said change replacing said non-delayed value with said delayed value as said processing stage output signal.

32. *(Original)* A method as claimed in claim 31, wherein supply of said delayed value to said following processing stage forces forward progress through processing operations.

33. *(Original)* A method as claimed in claim 25, wherein upon detection of said change forcing said delayed value to be stored in said non-delayed latch in place of said non-delayed value.

34. *(Original)* A method as claimed in claim 25, wherein processing operations within said processing stage and said following processing stage are driven by a non-delayed clock signal.

35. *(Original)* A method as claimed in claim 34, wherein upon detection of said change gating said non-delayed clock signal to provide time for said following processing stage to recover from input of said non-delayed value and instead use said delayed value.

36. *(Original)* A method as claimed in claim 35, wherein said non-delayed capture time is derived from a predetermined phase point of said non-delayed clock signal, a phased delayed version of said non-delayed clock signal is used as a delayed clock signal and said delayed capture time is derived from a predetermined phase point of said delayed clock signal.

37. *(Original)* A method as claimed in claim 25, wherein said plurality of processing stages are respective pipeline stages within a synchronous pipeline.

38. *(Original)* A method as claimed in claim 25, wherein a minimum processing time taken for said processing operation is greater than a time separating said delayed capture time from said non-delayed capture time such that said delayed value is not influenced by a processing operation performed upon different input values.

39. *(Original)* A method as claimed in claim 38, wherein said processing logic includes one or more delay elements to ensure said minimum processing time is exceeded.

40. *(Original)* A method as claimed in claim 25, wherein a maximum processing time taken for said processing operation is less than a sum of a time separating said delayed capture time from said non-delayed capture time and a time between non-delayed capture times such that said processing logic will have completed said processing operation by said delayed capture time.

41. *(Original)* A method as claimed in claim 25, wherein said processing stages are part of a data processor.

42. *(Original)* A method as claimed in claim 25, comprising storing a count of detection of errors corresponding to said change.

43. *(Currently Amended)* A method as claimed in claim ~~38~~42, wherein said count may be read by software.

44. *(Currently Amended)* A method as claimed in claim ~~29~~25, comprising monitoring work quantities including a quantity of useful work performed in progressing said processing operation and a quantity of work used to perform said error-recovery operations.

45. *(Original)* A method as claimed in claim 44, wherein one or more operating parameters are controlled in dependence upon said work quantities.

46. *(Original)* A method as claimed in claim 45, wherein said one or more operating parameters include at least one of:

an operating voltage;

an operating frequency;

an integrated circuit body bias voltage; and

temperature.

47. *(Original)* A method as claimed in claim 25, wherein said delayed latch also serves as a serial scan chain latch within a serial scan chain.

48. *(Currently Amended)* A method as claimed in claim ~~33~~25, wherein said delayed latch also serves as a serial scan chain latch within a serial scan chain and a signal value serially scanned in to said serial scan chain latch is forced in to said non-delayed latch during diagnostic operations using said error repair logic.